

MATERIA: PROYECTO INTEGRADOR

PROFESOR : GONZALO VERA

ALUMNO: RIOS LIONEL

AÑO: 2024

ACTIVIDADES

Informe de Aprendizaje: Escribir un resumen de lo aprendido en el tutorial.

¿Qué es Git y GitHub?

Git es un sistema de control de versiones de los archivos creados. Git realiza seguimiento de cambios, llevando un registro de lo trabajado posibilitando volver a una versión anterior.

GitHub, es Git en línea, o repositorio en la nube. Facilitando el trabajo con grupos de personas trabajando todas sobre un mismo proyecto.

¿Cuales son los comandos de Git y GitHub?

Cualquiera de las dos herramientas, posee o utiliza los mismos comandos.

Comandos para configuración de repositorio

Git config (primeramente configuramos nombre y mail)

Git init (inicializa el repositorio, creando un subdirectorio .git)

Git clone (clonar un repositorio local o remoto)

Comandos para cambios

Git add . (mueve los cambios realizados en el directorio local a la rama)

Git commit (confirma los cambios, y agrega un mensaje de aviso)

Git status (muestra si existe algún cambio o archivo sin seguimiento)

Git diff (para comparar dos conjunto de datos, se usa junto a git status y git log)

Git stash (almacena temporalmente los cambios sin confirmar)

Comandos examen repositorio

Git tag (crea una nueva etiqueta)

Git blame (Visualizar los metadatos de Autor adjuntos)

Comandos para deshacer cambios

Git clean (para deshacer cambios de archivos sin seguimiento)

Git revert (invierte los cambios realizados al confirmar seguimiento)

Git reset (deshace cambios, de 3 formas distintas soft, mixed y hard)

Git rm (elimina archivos sueltos o dejar de seguir algún archivo)

Comandos creación de ramas y unificación

Git Branch (son las ramas de trabajo)

Git Merge (permite tomar las ramas o líneas, y unificarlas)

Git Checkout (Permite ver los cambios de las ramas creadas por Git branch)

Comandos sincronización remota

Git Fetch (descarga confirmaciones, archivos y referencias de un repositorio remoto a tu repositorio local)

Git Push (Cargar contenido de un repositorio local a uno remoto)

Git Pull (traer y combinar repositorios remotos, reorganizandolos)

```
Lionel@kronnos MINGW64 ~/Documents/TEC.SUPERIOR TELECOMUNICACIONES ISPC/2024/PI1
/Primer semana/MrpP1_Rios_Lionel_grupo2 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Primer Proyecto VS/
    README.md

nothing added to commit but untracked files present (use "git add" to track)

Lionel@kronnos MINGW64 ~/Documents/TEC.SUPERIOR TELECOMUNICACIONES ISPC/2024/PI1
/Primer semana/MrpP1_Rios_Lionel_grupo2 (master)
$ git add .

Lionel@kronnos MINGW64 ~/Documents/TEC.SUPERIOR TELECOMUNICACIONES ISPC/2024/PI1
/Primer semana/MrpP1_Rios_Lionel_grupo2 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Primer Proyecto VS/Primer_proyecto_Plano
    new file:   README.md

Lionel@kronnos MINGW64 ~/Documents/TEC.SUPERIOR TELECOMUNICACIONES ISPC/2024/PI1
/Primer semana/MrpP1_Rios_Lionel_grupo2 (master)
$ git commit -m 'primer avance'
[master (root-commit) 5d40721] primer avance
 2 files changed, 3 insertions(+)
 create mode 100644 Primer Proyecto VS/Primer_proyecto_Plano
 create mode 100644 README.md

Lionel@kronnos MINGW64 ~/Documents/TEC.SUPERIOR TELECOMUNICACIONES ISPC/2024/PI1
/Primer semana/MrpP1_Rios_Lionel_grupo2 (master)
$ git status
On branch master
nothing to commit, working tree clean

Lionel@kronnos MINGW64 ~/Documents/TEC.SUPERIOR TELECOMUNICACIONES ISPC/2024/PI1
/Primer semana/MrpP1_Rios_Lionel_grupo2 (master)
$
```